

1、location

location 对象提供了与当前窗口中加载的文档有关的信息，还提供了一些导航的功能，它既是window对象的属性，也是document对象的属性

[JavaScript 秘密花园 \(bonsaiden.github.io\)](https://bonsaiden.github.io)

[JavaScript 教程-JavaScript中文网-JavaScript教程资源分享门户 \(javascriptcn.com\)](http://javascriptcn.com)

```
1 console.log(location);           //输出location对象
2 console.log(location.href);      //输出当前地址的全路径地址
3 console.log(location.origin);    //输出当前地址的来源
4 console.log(location.protocol);  //输出当前地址的协议
5 console.log(location.hostname);  //输出当前地址的主机名
6 console.log(location.host);      //输出当前地址的主机
7 console.log(location.port);      //输出当前地址的端口号
8 console.log(location.pathname);  //输出当前地址的路径部分
9 console.log(location.search);    //输出当前地址的?后边的参数部分
```

```
1 <script>
2     console.log(window.location);
3 </script>
```

```
▼ Location ⓘ
  ▶ ancestorOrigins: DOMStringList {length: 0}
  ▶ assign: f assign()
    hash: ""
    host: ""
    hostname: ""
    href: "file:///E:/Web%E5%89%8D%E7%AB%AF/J/day19/%E4%B8%8A%E8%AF%BE%E6%96%87%E6%A1%A3/02.19day.BOM.location.html"
    origin: "file://"
    pathname: "/E:/Web%E5%89%8D%E7%AB%AF/J/day19/%E4%B8%8A%E8%AF%BE%E6%96%87%E6%A1%A3/02.19day.BOM.location.html"
    port: ""
    protocol: "file:"
  ▶ reload: f reload()
  ▶ replace: f replace()
    search: ""
  ▶ toString: f toString()
  ▶ valueOf: f valueOf()
    Symbol(Symbol.toPrimitive): undefined
  ▶ [[Prototype]]: Location
```

1、location.href

- 返回当前加载页面的完整URL
- location.href 与 window.location.href等价
- href: 是一个属性，通过修改该属性值可以实现页面跳转，是可以后退的；

当location.href赋值时会跳转

```
1 setTimeout(function(){
2     window.location.href = './1.练习.html';
3 }, 3000);
```

2、location.hash

- 返回URL中的hash（#号后跟零或多个字符），如果不包含则返回空字符串

3、location.host

- 返回服务器名称和端口号

4、location.hostname

- 返回不带端口号的服务器名称

5、location.pathname

- 返回URL中的目录和（或）文件名

6、location.port

- 返回URL中指定的端口号，如果没有返回空字符串

7、location.protocol

- 返回页面使用的协议

8、location.search

- 返回URL的查询字符串，查询？开头的字符串

```
1 console.log(window.location);
2     window.location.querystring = function(){
3         var obj = {};
4         //this 就是当前对象window.location
5         this.search.substring(1).split('&').forEach(function(q){
6             obj[q.split('=').shift()] = q.split('=').pop();
7         });
8         return obj;
9     }
```

9、location.replace()

- 重新定向URL，不会在历史记录中生成新纪录（没有后退按钮）
- Location.replace() 方法以给定的URL来替换当前的资源。与assign() 方法不同的是，调用replace() 方法后，当前页面不会保存到会话历史中（session History），这样，用户点击回退按钮时，将不会再跳转到该页面。

```

2  setTimeout(function(){
3      //window.location.href = './1.倒计时.html';
4      window.location.replace('./1.倒计时.html');
5      //window.location.assign('./1.倒计时.html');
6  }, 3000);

```

10、location.reload()

- Location.reload() 方法用来刷新当前页面。该方法只有一个参数，当值为 true 时，将强制浏览器从服务器加载页面资源，当值为 false 或者未传参时，浏览器则可能从缓存中读取页面。

```

1  location.reload(true);
2  // 无缓存刷新页面（但页面引用的资源还是可能使用缓存，
3  // 大多数浏览器可以通过设置在打开开发者工具时禁用缓存实现无缓存需求）
4  console.log(Math.random());
5  setTimeout(function(){
6      window.location.reload();//刷新页面
7  }, 1000);

```

11、Location.assign()

- Location.assign()方法会触发窗口加载并显示指定的URL的内容

```

1
2  function newDoc(){
3      window.location.assign("http://www.w3cschool.cn")
4  }
5  </script>
6  </head>
7  <body>
8
9  <input type="button" value="加载新文档" onclick="newDoc()">
10
11 </body>

```

BOM.打开新的页面

```

1  setTimeout(function(){
2      //是会被浏览器拦截的，代码的正常执行完全看用户

```

```
3     window.open('http://baidu.com');
4 }, 1000);
```

2、Navigator对象

Navigator代表的当前浏览器的信息，通过该对象可以来识别不同的浏览器，

userAgent: "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36(KHTML, like Gecko) Chrome/100.0.4896.60 Safari/537.36"

```
1 //系统信息 UA
2 console.log(window.navigator);
```

读取文件

安全性：没有权限读取本地磁盘文件，除非授权；

```
1 <style>
2     .avatar {
3         display: block;
4         width: 120px;
5         height: 120px;
6         border-radius: 50%;
7         background-size: cover;
8         cursor: pointer;
9     }
10
11     .avatar img {
12         width: 100%;
13         height: 100%;
14         border-radius: 50%;
15     }
16
17     .hide {
18         display: none;
19     }
20 </style>
21 </head>
22
23 <body>
24     <label class="avatar" for="avatar">
```

```

25     
26 </label>
27 <input type="file" id="avatar" class="hide">
28 <script>
29     //当选择文件之后，内容发生改变，触发change事件
30     document.querySelector('input[type="file"]').onchange = function () {
31         console.dir(this.files[0]);
32         //创建一个文件读取的实例化对象
33         var fr = new FileReader();
34         fr.readAsDataURL(this.files[0]);
35         //xuyao等读取完成，就把读取的结果显示到img标签身上
36         fr.onload = function () {
37             document.querySelector('#show').src = this.result;
38         }
39     }
40 </script>

```

3、History对象

History对象可以用来操作浏览器向前或向后翻页

```

1 console.log(history);           //输出history对象
2 console.log(history.length);    //可以获取到当前访问的链接数量

```

1. back(): 可以回退到上一个页面，作用和浏览器的回退按钮一样

```
1 history.back();
```

2. forward(): 可以跳转到下一个页面，作用和浏览器的前进按钮一样

```
1 history.forward();
```

3. go(): 可以用来跳转到指定的页面，它需要一个整数作为参数

- 1: 表示向前跳转一个页面，相当于forward()
- 2: 表示向前跳转两个页面
- -1: 表示向后跳转一个页面，相当于back()
- -2: 表示向后跳转两个页面

```
1 history.go(-1);
```

4、Screen对象

1属性

属性	描述
<code>availHeight</code>	返回显示屏幕的高度 (除 Windows 任务栏之外)。
<code>availWidth</code>	返回显示屏幕的宽度 (除 Windows 任务栏之外)。
<code>bufferDepth</code>	设置或返回调色板的比特深度。
<code>colorDepth</code>	返回目标设备或缓冲器上的调色板的比特深度。
<code>deviceXDPI</code>	返回显示屏幕的每英寸水平点数。
<code>deviceYDPI</code>	返回显示屏幕的每英寸垂直点数。
<code>fontSmoothingEnabled</code>	返回用户是否在显示控制面板中启用了字体平滑。
<code>height</code>	返回显示屏幕的高度。
<code>logicalXDPI</code>	返回显示屏幕每英寸的水平方向的常规点数。
<code>logicalYDPI</code>	返回显示屏幕每英寸的垂直方向的常规点数。
<code>pixelDepth</code>	返回显示屏幕的颜色分辨率 (比特每像素) 。
<code>updateInterval</code>	设置或返回屏幕的刷新率。
<code>width</code>	返回显示器屏幕的宽度。

5、常用javascript跳转方法

第一种:

```
1 <script>
2     window.location.href = "http://www.baidu.com";
3 </script>
```

第二种:

```
1 <script>
2     alert("返回");
```

```
3     window.history.back(-1);
4 </script>
```

第三种： 这个方法是只针对IE的，不适用于其他浏览器

```
1 <script>
2     window.navigate("http://www.baidu.com");
3     //这个方法是只针对IE的，不适用于其他浏览器
4 </script>
```

第四种：

```
1 <script>
2     self.location = 'http://www.baidu.com';
3 </script>
```

第五种：

```
1 <script>
2     alert("非法访问！");
3     top.location = 'http://www.baidu.com';
4 </script>
```